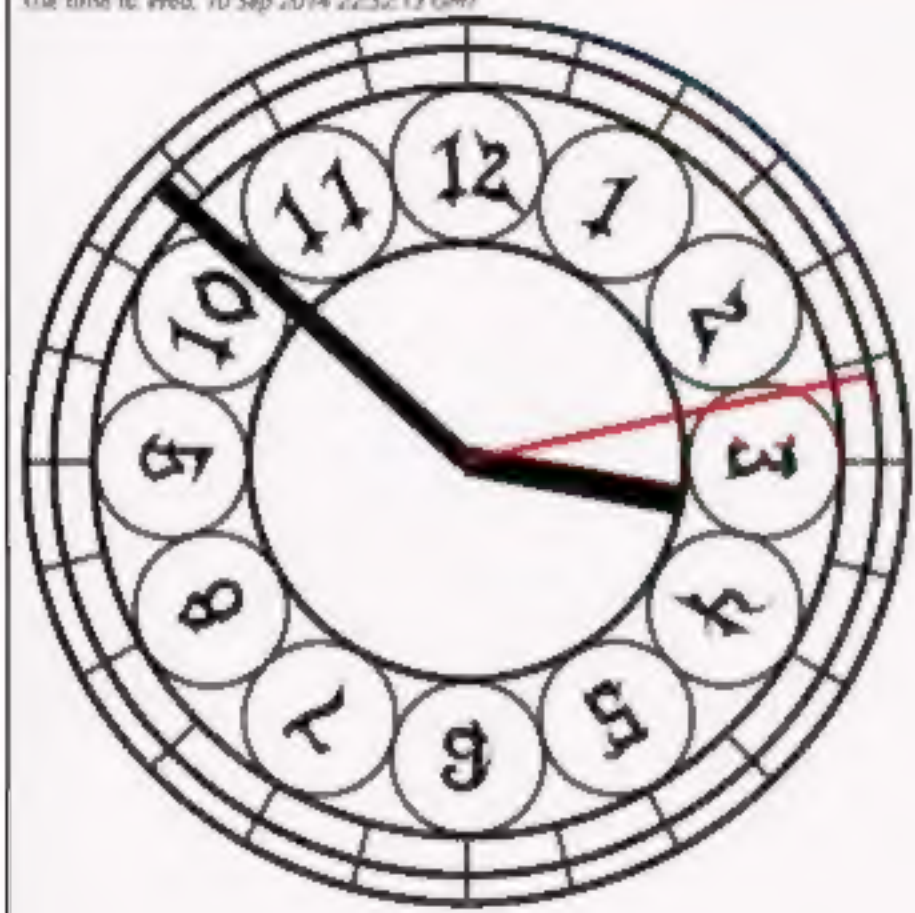
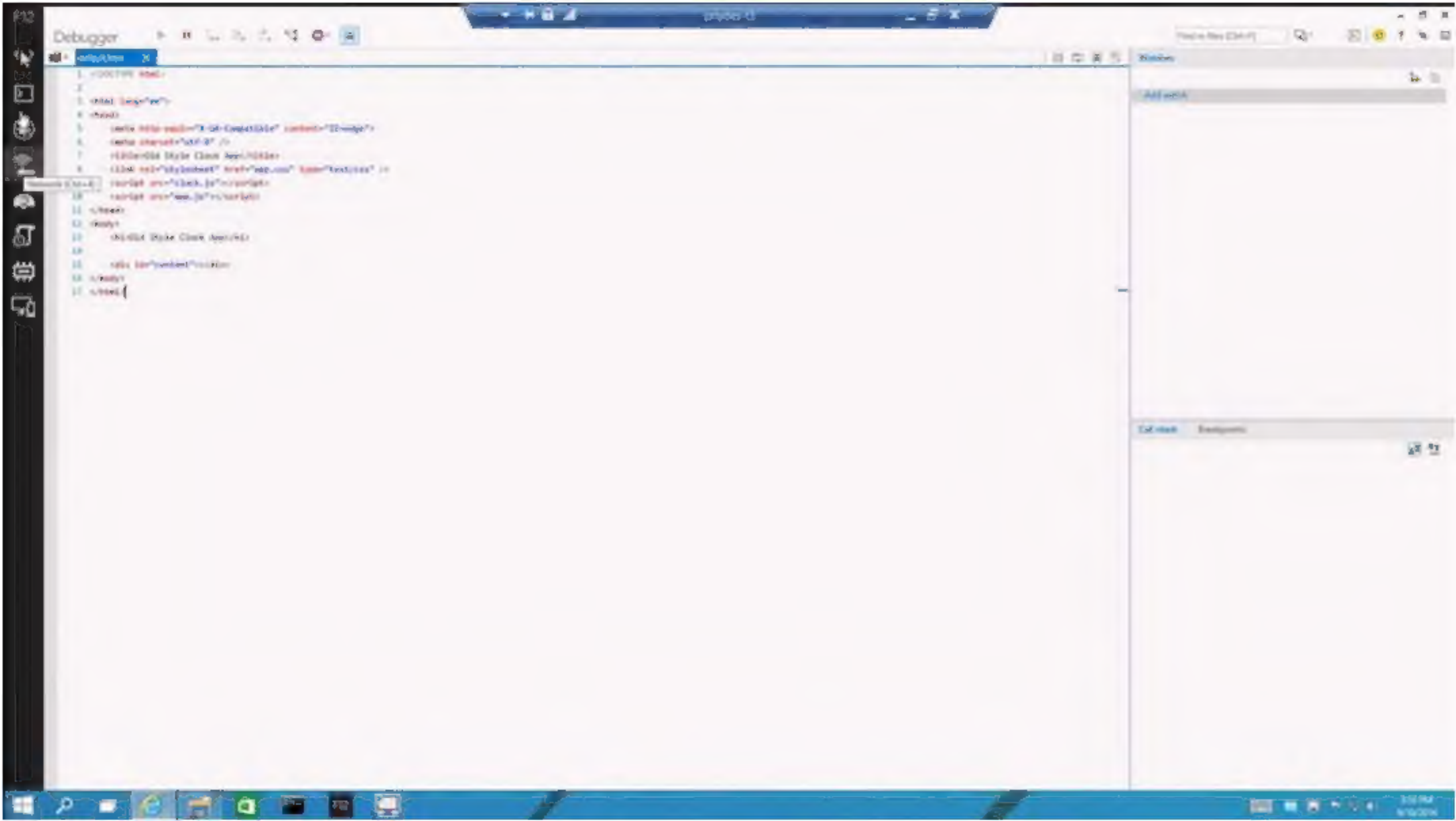


# Old Style Clock App

The time is: Wed, 10 Sep 2014 22:52:12 GMT





Debugger | Files | Code | **code**

```
1 // Initialize jQuery, jQuery, etc.
2
3 module Click {
4   export class App {
5     element: HTMLDivElement;
6     span: HTMLSpanElement;
7     timerSpan: number;
8     click: Click.ClickEvent;
9
10    constructor(element: HTMLDivElement) {
11      this.element = element;
12      this.element.innerHTML += "The click is ";
13      this.span = document.createElement("span");
14      this.element.appendChild(this.span);
15      this.span.innerHTML = new Date().toLocaleString();
16
17      var clockId = document.createElement("div");
18      element.appendChild(clockId);
19      this.click = new Click.ClickEvent();
20    }
21
22    start() {
23      this.timerSpan = setInterval(() => { this.span.innerHTML = new Date().toLocaleString(); }, 1000);
24    }
25
26    stop() {
27      clearInterval(this.timerSpan);
28    }
29  }
30 }
31
32 // Main module
33 export module Main {
34   export function main() {
35     var app = document.getElementsByTagName("div")[0];
36     var greater = new Click.App(app);
37     greater.start();
38   }
39 }
```

Watch

Global

app	[object App]
arguments	[object Arguments]
element	[object HTMLDivElement]
span	[object HTMLSpanElement]
timerSpan	[object HTMLDivElement]
click	[object Click.ClickEvent]
clockId	[object HTMLDivElement]
main	[object Function]

Call stack

Breakpoints

Main module

App	app (7) 10
main	app (7) 10



2000

+ [View all posts by](#) [Jesse Cougle](#)

- ☐ ~~the~~
- ☐ ~~equivalent~~
- ☐ ~~element~~
- ☐ ~~shortly~~
- ☐ (drop)
- ☐ (eliminate)

- [Topic] App
- [Topic] Algorithms
- [Topic] OOPS/Full Course
- [Topic] OOPS/Full Course
- [Topic] Full Course

Cellulose	Starch
<p>Cellulose is a polysaccharide made of glucose units linked by <math>\beta</math>-1,4 glycosidic bonds. It is the most abundant organic polymer on Earth. It is a linear chain of glucose units. It is insoluble in water. It is a structural component of plant cell walls. It is a major component of paper and cardboard. It is a major component of cotton and linen. It is a major component of wood. It is a major component of paper and cardboard. It is a major component of cotton and linen. It is a major component of wood.</p>	<p>Starch is a polysaccharide made of glucose units linked by <math>\alpha</math>-1,4 glycosidic bonds. It is the most abundant carbohydrate in the world. It is a linear chain of glucose units. It is soluble in water. It is a storage form of glucose in plants. It is a major component of food. It is a major component of paper and cardboard. It is a major component of cotton and linen. It is a major component of wood.</p>

100

App	app (4.1)
services	services (1.1)

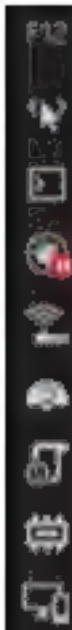












```
1 module Clock {
2   export class ClockFace {
3     private static IP: string = "https://www.ck12.org/";
4     private static MinuteIncrement: number = 5;
5     private static HourIncrement: number = 10;
6
7     private element: HTMLDivElement;
8     private hourHand: HTMLDivElement;
9     private minuteHand: HTMLDivElement;
10    private secondHand: HTMLDivElement;
11
12    constructor(element: HTMLDivElement) {
13      var image: HTMLImageElement = document.createElement("img");
14      image.src = ClockFace.IP;
15
16      element.appendChild(image);
17      element.appendChild(image);
18
19      // Create hands
20      this.hourHand = document.createElement("div");
21      this.hourHand.className = "hourHand";
22      element.appendChild(this.hourHand);
23
24      this.minuteHand = document.createElement("div");
25      this.minuteHand.className = "minuteHand";
26      element.appendChild(this.minuteHand);
27
28      this.secondHand = document.createElement("div");
29      this.secondHand.className = "secondHand";
30      element.appendChild(this.secondHand);
31
32      setInterval(() => {
33        var date = new Date();
34
35        var hours = date.getHours() % 12;
36        var minutes = date.getMinutes();
37        var seconds = date.getSeconds();
38
39        this.hourHand.style.transform = `rotate(${(ClockFace.HourIncrement / 12) * hours}deg)`;
40        this.minuteHand.style.transform = `rotate(${(ClockFace.MinuteIncrement / 60) * minutes}deg)`;
41        this.secondHand.style.transform = `rotate(${(ClockFace.MinuteIncrement / 60) * seconds}deg)`;
42      }, 1000);
43    }
44  }
45 }
46
47 }
```

Variable	Value
IP	https://www.ck12.org/
MinuteIncrement	5
HourIncrement	10
element	HTMLDivElement
hourHand	HTMLDivElement
minuteHand	HTMLDivElement
secondHand	HTMLDivElement
constructor	function
setInterval	function
new Date	Date
getHours	function
getMinutes	function
getSeconds	function
style	Object
transform	string

Function	File	Line
----------	------	------

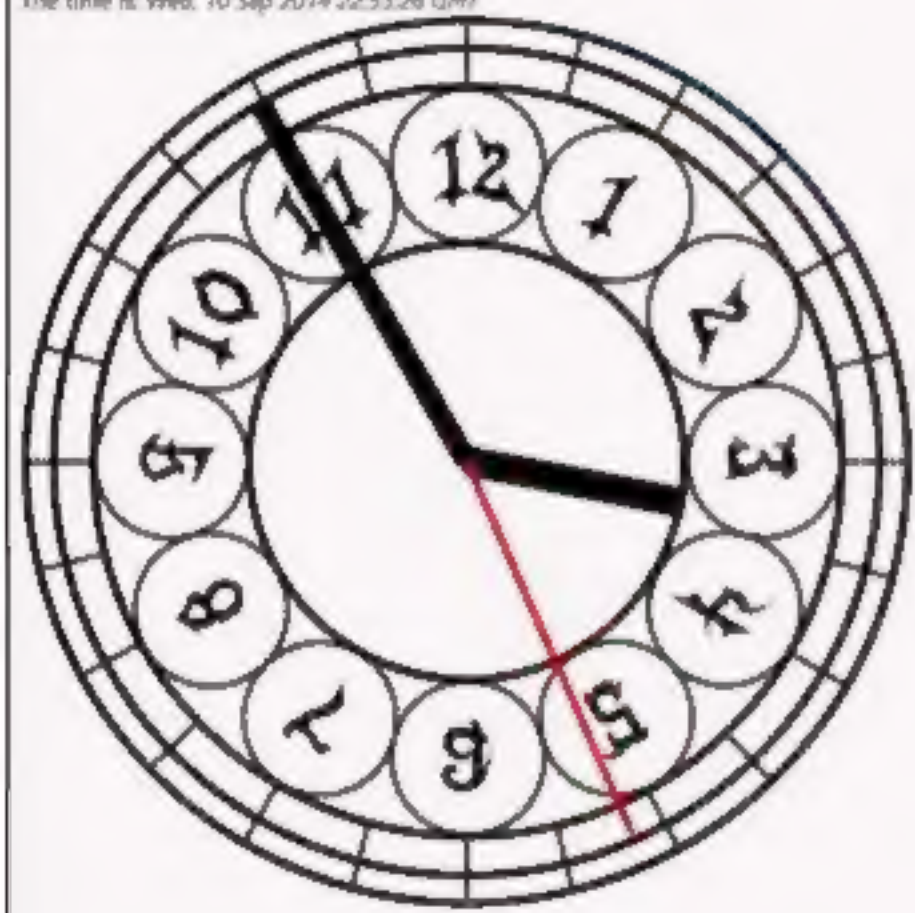
Anonymous function	Clock.js	15
--------------------	----------	----

Anonymous function	Clock.js	15
--------------------	----------	----



## Old Style Clock App

The time is: Wed, 10 Sep 2014 22:55:26 GMT

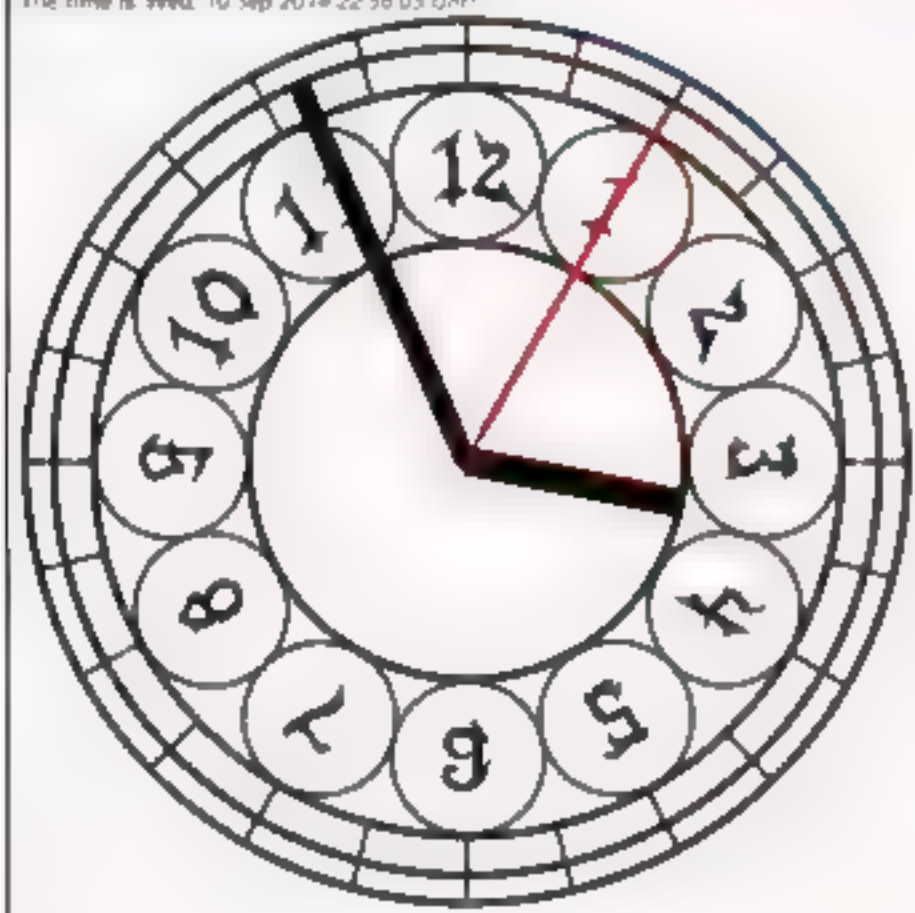


```
1 // constructor for ClockApp
2
3 module Clock {
4     export class App {
5         element: HTMLElement;
6         span: HTMLElement;
7         timerId: number;
8         clock: Clock.ClockFace;
9
10         constructor(element: HTMLElement) {
11             this.element = element;
12             this.element.innerHTML = `The time is `;
13             this.span = document.createElement('span');
14             this.element.appendChild(this.span);
15             this.span.innerHTML = new Date().toLocaleTimeString();
16
17             var clockDiv = document.createElement('div');
18             element.appendChild(clockDiv);
19             this.clock = new ClockFace(clockDiv);
20         }
21
22         start() {
23             this.timerId = setInterval(() => this.span.innerHTML = new Date().toLocaleTimeString(), 1000);
24         }
25
26         stop() {
27             clearInterval(this.timerId);
28         }
29     }
30 }
31
32 // app.js
33 window.onload = () => {
34     var el = document.getElementsByTagName('body')[0];
35     var greater = new Clock.App(el);
36     greater.start();
37 }
```



# Old Style Clock App

The time is: Wed, 10 Sep 2014 22:56:05 GMT





```
def main():
    # Create a list of numbers
    numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

    # Iterate over the list and print each number
    for number in numbers:
        print(number)

    # Calculate the sum of the numbers
    total = 0
    for number in numbers:
        total += number

    # Print the total
    print(total)
```

File Explorer

Address bar: C:\Users\user\Documents

Files and folders: document1.docx, document2.docx, document3.docx

Task View

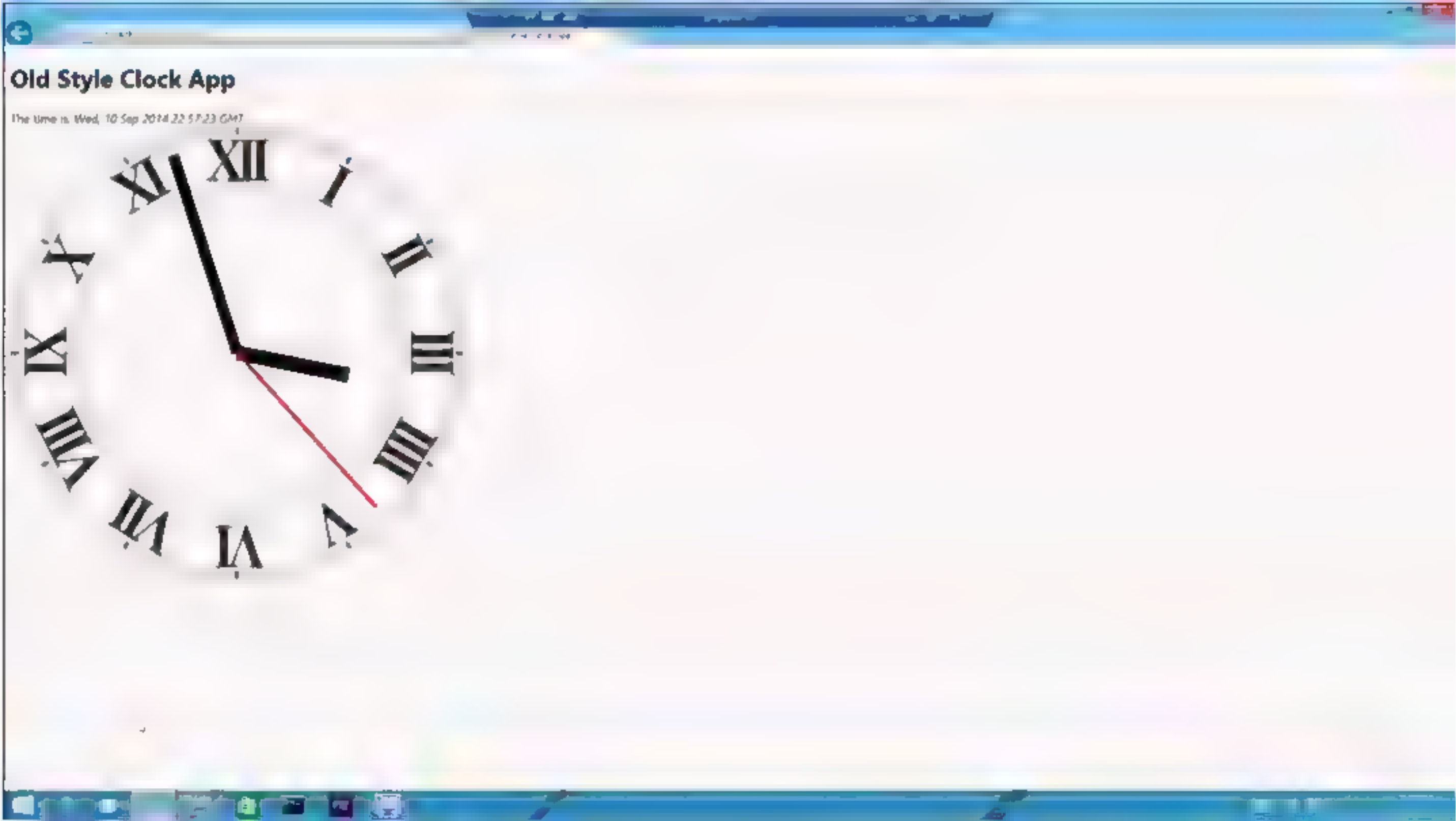
Taskbar: task1, task2, task3

System Tray

System clock: 10:00 AM, 10/10/2023







## Old Style Clock App

The time is: Wed, 10 Sep 2014 22:57:23 GMT







Visual Studio Code interface showing a C# project named "Project1" in the Solution Explorer on the left. The main editor displays a file named "Program.cs" with the following code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Project1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
        }
    }
}
```

The right sidebar shows the "Properties" window for the selected file, displaying various attributes such as Name, Extension, and Date Modified.



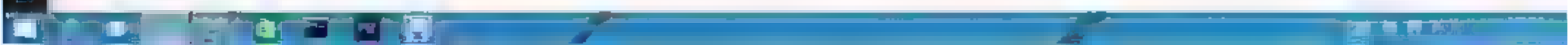


Microsoft Word document interface showing a blank page with a ribbon at the top and a sidebar on the right.

**Ribbon (Top):** The ribbon is visible, showing the 'File' tab and the 'Home' tab. The 'Home' tab includes sections for Font, Paragraph, Styles, and Editing.

**Sidebar (Right):** The sidebar is open, displaying the 'Navigation' pane. It shows a list of sections in the document, including 'Section 1' and 'Section 2'. The 'Section 1' is currently selected.

**Document Content:** The main body of the document is blank, with a few faint lines of text visible at the top, likely from a previous page or a header.

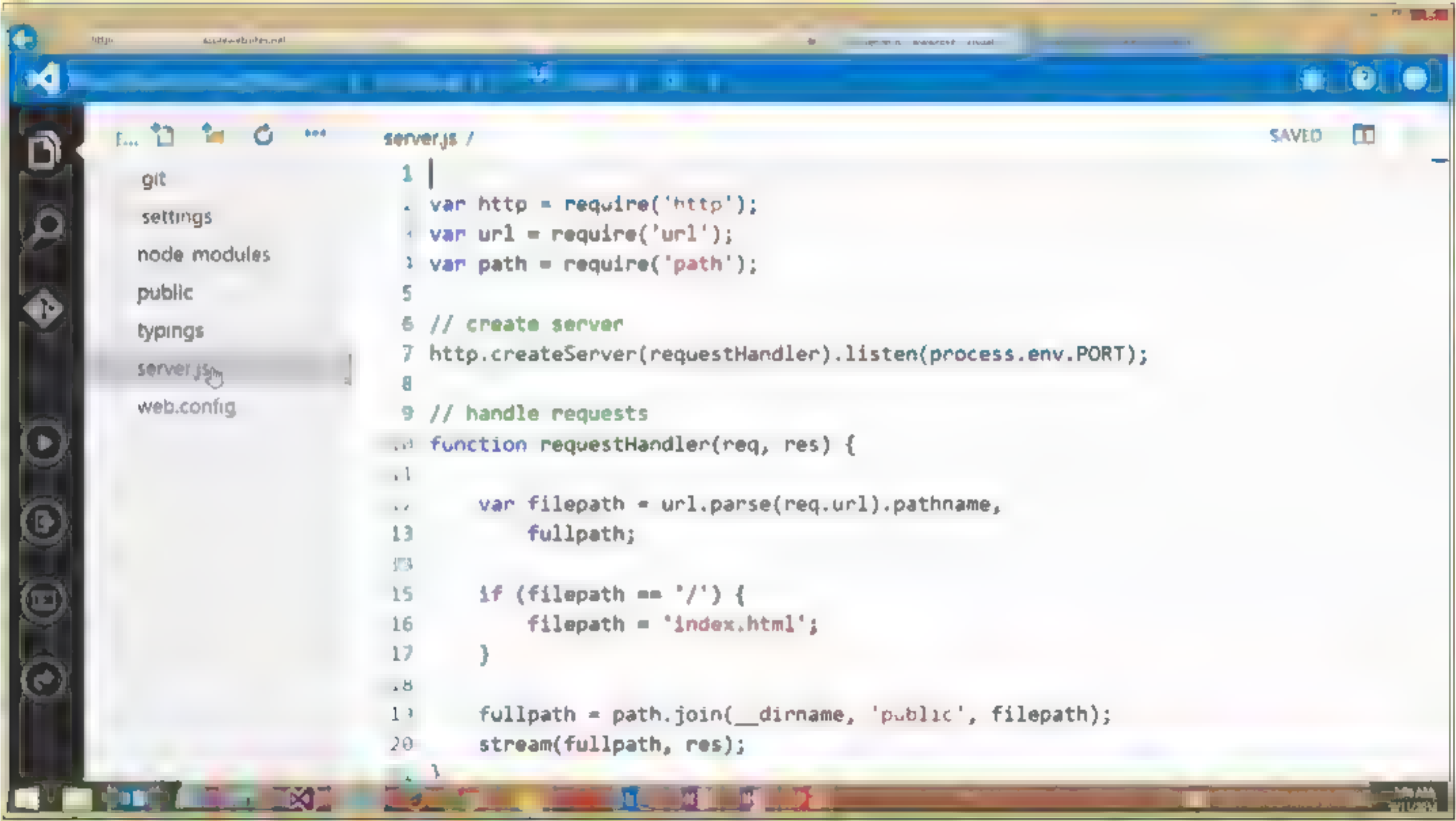




# TypeScript – the man behind the curtain of the Monaco JavaScript tools

Erich Gamma, Johannes Rieken  
VS Platform Tools/Monaco

»Chuck Norris can  
solve the Towers of  
Hanoi in one move.«



server.js /

SAVED

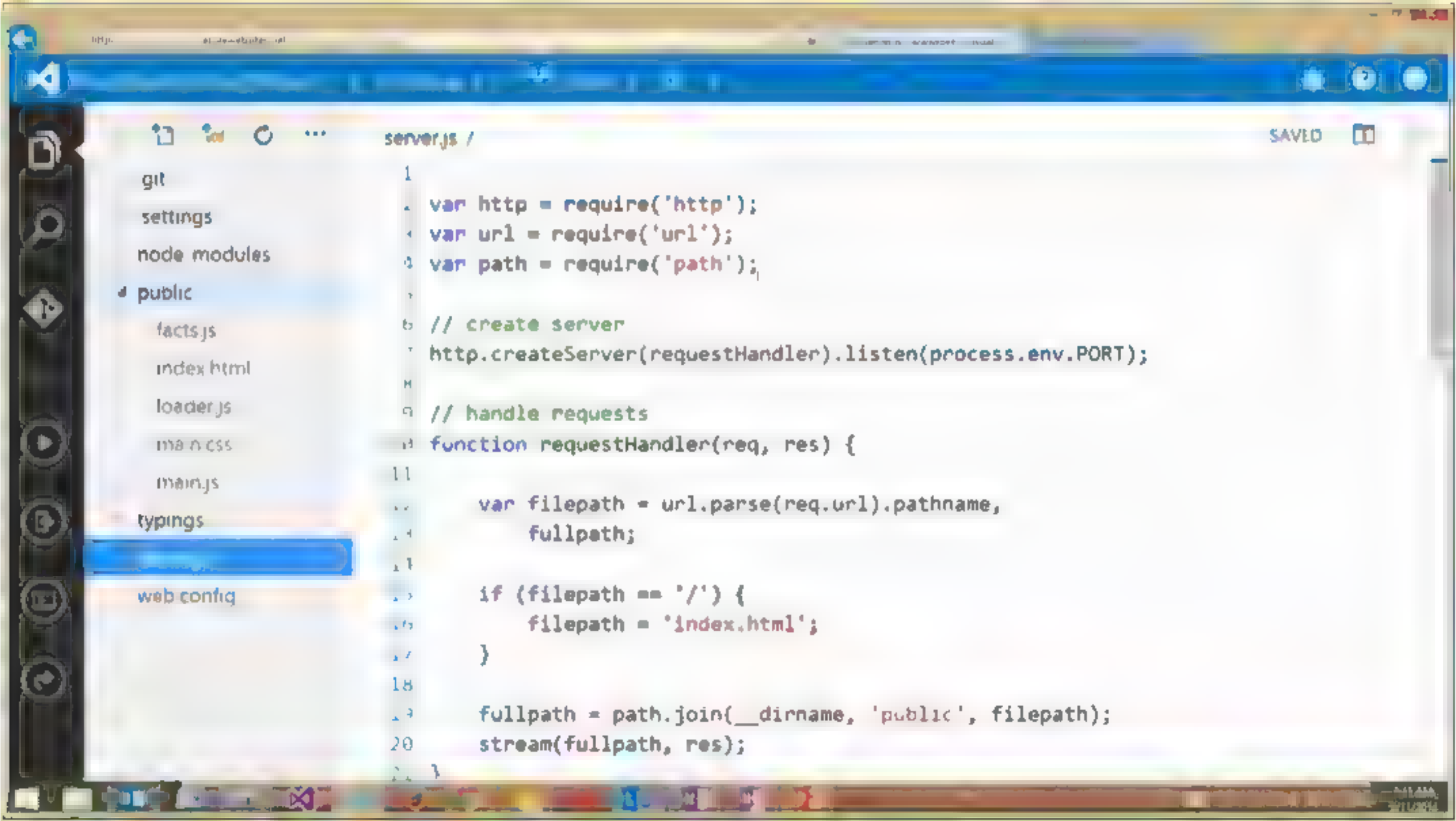
```
1 |
2 var http = require('http');
3 var url = require('url');
4 var path = require('path');
5
6 // create server
7 http.createServer(requestHandler).listen(process.env.PORT);
8
9 // handle requests
10 function requestHandler(req, res) {
11
12     var filepath = url.parse(req.url).pathname,
13         fullpath;
14
15     if (filepath == '/') {
16         filepath = 'index.html';
17     }
18
19     fullpath = path.join(__dirname, 'public', filepath);
20     stream(fullpath, res);
21 }
```

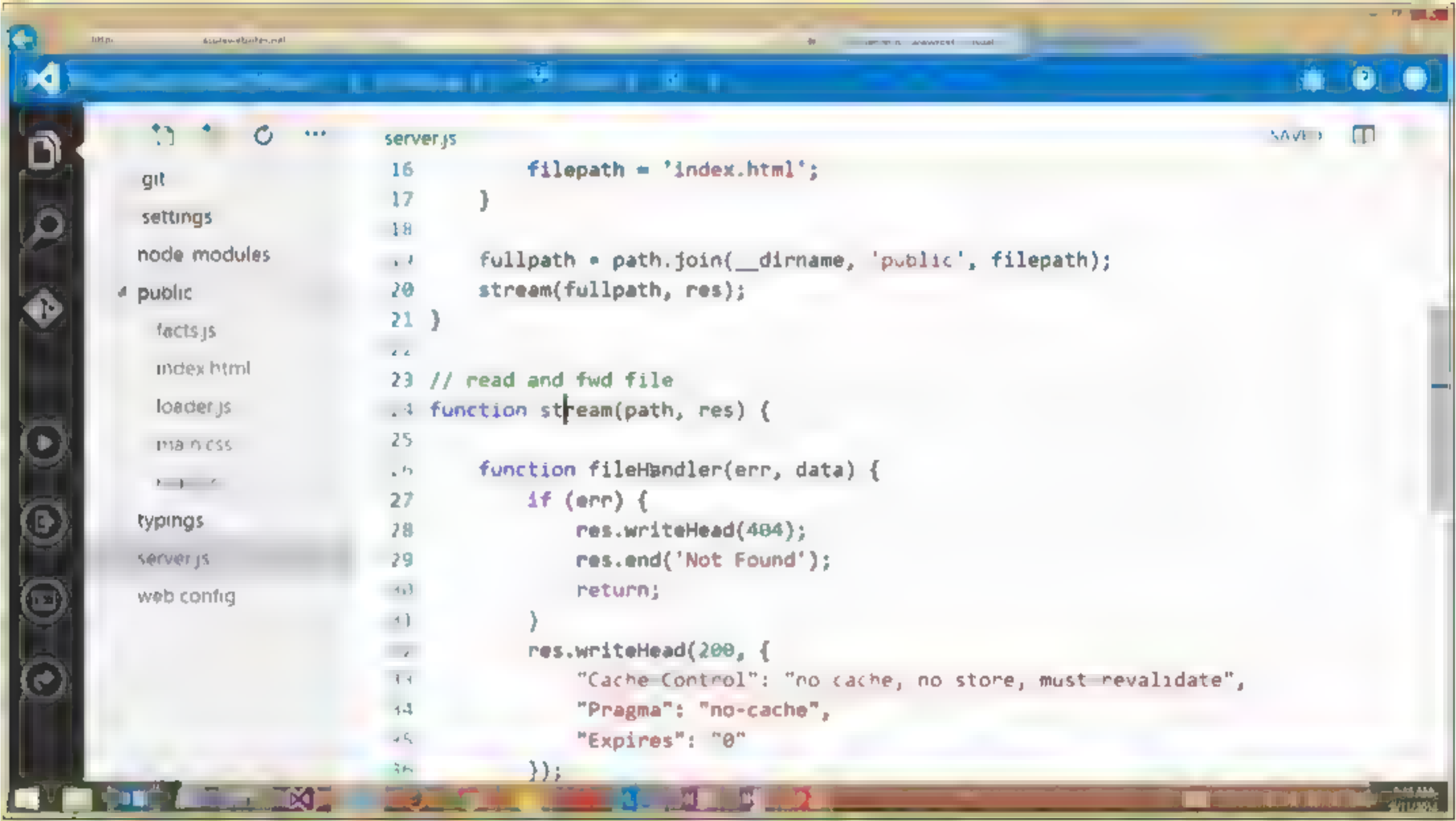
»Chuck Norris doesn't  
use web standards as  
the web will conform  
to him.«



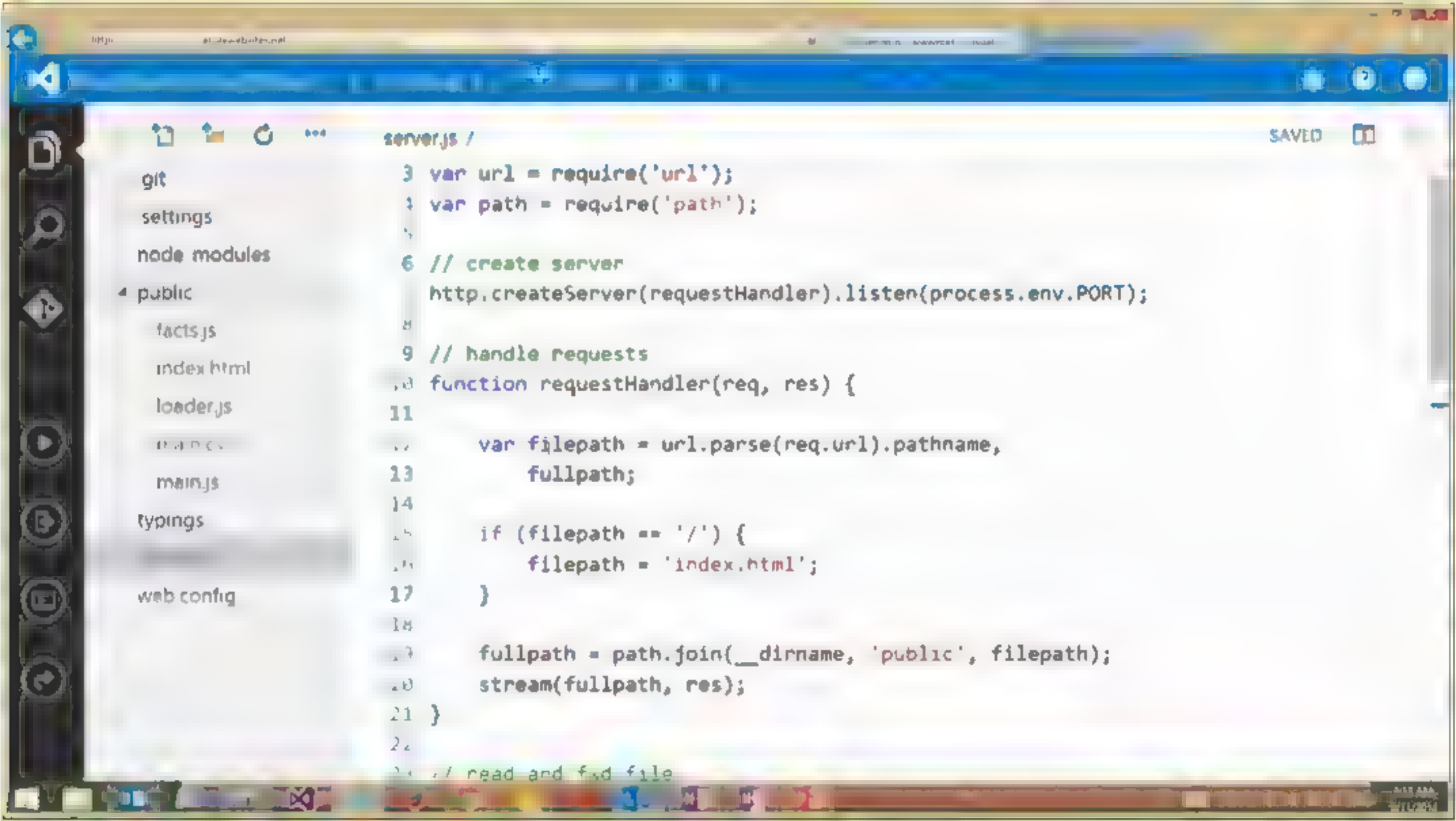
» Whiteboards are  
white because Chuck  
Norris scared them  
that way.«

»Chuck Norris burst  
the dot com bubble.«









server.js

SAVED



```
1
2 var http = require('http');
3 var url = require('url');
4 var path = require('path');
5
6 // create server
7 http.createServer(requestHandler).listen(process.e
8
9 // handle requests
10 function requestHandler(req, res) {
11
12     var filepath = url.parse(req.url).pathname,
13         fullpath;
14
15     if (filepath == '/') {
16         filepath = 'index.html';
17     }
18
19     fullpath = path.join(__dirname, 'public', file
20     stream(fullpath, res);
```

```
59 "baseUrl": "",
60 "lint": {
61     "undeclaredVar Use '===' and '===' instead of '==' and
62     "comparisonOperatorsNotStrict": "ignore",
63     "missingSemicolon": "ignore",
64     "curlyBracketsMustNotBeOmitted": "ignore",
65     "newOnLowercaseFunctions": "warning",
66     "newOnReturningFunctions": "warning",
67     "typeScriptSpecifics": "error",
68     "emptyBlocksWithoutComment": "ignore",
69     "reservedKeywords": "warning",
70     "unknownTypeOfResults": "warning",
71     "semicolonsInsteadOfBlocks": "ignore",
72     "functionsInsideLoops": "ignore",
73     "tripleSlashReferenceAlike": "warning",
74     "unusedVariables": "ignore",
75     "unusedFunctions": "ignore",
76     "redeclaredVariables": "ignore",
77     "unknownProperty": "warning",
78     "primitivesInInstanceOf": "error",
```

server.js

```
1
2 var http = require('http');
3 var url = require('url');
4 var path = require('path');
5
6 // create server
7 http.createServer(requestHandler).listen(process.env.PORT);
8
9 // handle requests
10 function requestHandler(req, res) {
11
12     var filepath = url.parse(req.url).pathname,
13         fullpath;
14
15     if (filepath === '/') {
16         filepath = 'index.html';
17     }
18
19     fullpath = path.join(__dirname, 'public', filepath);
20     stream(fullpath, res);
21 }
```

server.js

```
1 /*  
2  */  
3 var http = require('http');  
4 var url = require('url');  
5 var path = require('path');  
6  
7 // create server  
8 http.createServer(requestHandler).listen(process.env.PORT),  
9  
10 // handle requests  
11  
12 function requestHandler(req, res) {  
13  
14     var filepath = url.parse(req.url).pathname,  
15     fullpath;  
16  
17     if (filepath === '/') {  
18         filepath = 'index.html',  
19     }  
20  
21     fullpath = path.join(__dirname, 'public', filepath),  
22     stream(fullpath, res);  
23 }  
24
```

server.js

SAVIN

```
1 ref |
2   msCancelRequestAnimationFrame
3   var msRequestAnimationFrame
4   var reference
5   var Relative Reference to another File
6   ReferenceError
7   // requestAnimationFrame
8   http SVGURIReference
9
10 //
11 fun
12
13
14
15
16 if (filepath === '/') {
17     filepath = 'index.html';
18 }
19
20 fullpath = path.join(__dirname, 'public', filepath);
21 stream(fullpath, res);
```



server.js

SAVED

```
1      "typings node/node.d.ts"
2  Invalid 'reference' directive syntax.
3  Did you mean '/// <reference path=\"some/path.ts\" />'?
4  var http = require('http');
5  var url = require('url');
6  var path = require('path');
7
8  // create server
9  http.createServer(requestHandler).listen(process.env.PORT);
10
11 // handle requests
12 function requestHandler(req, res) {
13
14     var filepath = url.parse(req.url).pathname,
15         fullpath;
16
17     if (filepath === '/') {
18         filepath = 'index.html';
19     }
20
21     fulloath = path.join( dirname, 'public', filepath);
```

server.js

```
9 http.createServer(requestHandler).listen(process.env.PORT);
10
11 // handle requests
12 function requestHandler(req, res) {
13
14     var filepath = url.parse(req.url).pathname,
15         fullpath;
16
17     if (filepath === '/') {
18         filepath = 'index.html';
19     }
20
21     fullpath = path.join(__dirname, 'public', filepath);
22     stream(fullpath, res);
23 }
24
25 // read and fwd file
26 function stream(path, res) {
27
28     function fileHandler(err, data) {
29         if (err) {
```

server.js

```
4 var http = require('http');
5 var url = require('url');
6 var path = require('path');
7
8 // create server
9 http.createServer(requestHandler).listen(process.env.PORT);
10
11 // handle requests
12 function requestHandler(req, res) {
13
14     var filepath = url.parse(req.url).pathname,
15         fullpath;
16
17     if (filepath === '/') {
18         filepath = 'index.html';
19     }
20
21     fullpath = path.join(__dirname, 'public', filepath);
22     stream(fullpath, res);
23 }
```

server.js

SAVE

```
1      __"typings/node/node.d.ts"
2
3
4  var http = require('http');
5  var url = require('url');
6  var path = require('path');
7
8  // create server
9  http.createServer(requestHandler).listen(process.env.PORT);
10
11 // handle requests
12 function requestHandler(req, res) {
13
14     var filepath = url.parse(req.url).pathname,
15         fullpath;
16
17     if (filepath === '/') {
18         filepath = 'index.html';
19     }
20
21     fulloath = path.join( dirname, 'public', filepath);
```

```

11 fullpath = path.join(__dirname, public, filepath),
22 stream(fullpath, res);
23 }
24
25 read and find file
26 function stream path, res, {
27
28   function fileHandler(err, data) {
29     if (err) {
30       res.writeHead(404),
31       res.end('Not Found');
32       return;
33     }
34     res.writeHead(200, {
35       "Cache-Control": "no-cache, no-store, must-revalidate",
36       "Pragma": "no-cache",
37       "Expires": "0"
38     });
39     res.write(data.toString());
40     res.end();
41   });
42
43   require( fs ) readFile path, fileHandler,
44 }

```

```

1 // ping node node d ts
2
3
4 declare var exports any, declare var module any, declare var
5 import http = require('http');
6 import url = require( url ),
7 import path = require( path ),
8 import fs = require('fs');
9
10 var http = http,
11 var url = url,
12 var path = path,
13
14 // create server
15 http.createServer(requestHandler) listen(process.env.PORT),
16
17 handle requests
18 function requestHandler(req, res) {
19
20   var filepath = url.parse(req.url).pathname,
21   fullpath,
22
23   if (filepath === '/') {
24     filepath = 'index.html';
25   }
26
27   fullpath = path.join(__dirname, public, filepath),
28

```



```

1      typings/node/node.d.ts
2
3
4  var http = require('http');
5  var url = require('url');
6  var path = require('path');
7
8  // create server
9  http.createServer(requestHandler).listen(process.env.PORT),
10
11 // handle requests
12 function requestHandler(req, res) {
13
14     var filepath = url.parse(req.url).pathname,
15         fullpath;
16
17     if (filepath === '/') {
18         filepath = 'index.html';
19     }
20
21     fullpath = path.join(__dirname, public, filepath),
22     stream(fullpath, res);
23 }
24
25 // read and fwd file
26 function stream(path, res) {

```

```

1      typings/node/node.d.ts" />
2
3
4  declare var exports any, declare var module any, declare va
5  import http = require('http');
6  import url = require('url');
7  import path = require('path');
8  import fs = require('fs');
9  var http = http,
10  var url = url;
11  var path = path,
12
13 // create server
14 http.createServer(requestHandler).listen(process.env.PORT),
15
16 // handle requests
17 function requestHandler(req, res) {
18
19     var filepath = url.parse(req.url).pathname,
20         fullpath,
21
22     if (filepath === '/') {
23         filepath = 'index.html';
24     }
25
26     fullpath = path.join(__dirname, public, filepath),

```

```

1      typings node node d ts
2
3
4      var http = require( 'http' ),
5      var url = require( 'url' );
6      var path = require( 'path' );
7
8      // create server
9      http.createServer(requestHandler).listen(process.env PORT);
10
11     // handle requests
12     function requestHandler(req, res) {
13
14         var filepath = url.parse(req.url).pathname,
15             fullpath,
16
17         if (filepath === '/') {
18             filepath = 'index.html',
19         }
20
21         fullpath = path.join(__dirname, 'public', filepath),
22         stream(fullpath, res),
23     }
24
25     read and find file
26     function stream(path, res) {

```

```

4      import url = require('url'),
5      import path = require('path');
6      import fs = require('fs');
7
8      var http = http,
9      var url = url,
10     var path = path,
11
12
13     // create server
14     http.createServer(requestHandler).listen(process.env PORT);
15
16     // handle requests
17     function requestHandler(req, res) {
18
19         var filepath = url.parse(req.url).pathname,
20             fullpath,
21
22         if (filepath === '/') {
23             filepath = 'index.html',
24         }
25
26         fullpath = path.join(__dirname, 'public', filepath),
27         stream(fullpath, res),
28     }
29
30     read and find file
31     function stream(path, res) {

```

```

1      t, pings node node d ts
2
3
4  var http = require('http');
5  var url = require('url');
6  var path = require('path');
7
8  // create server
9  http.createServer(requestHandler).listen(process.env.PORT);
10
11 // handle requests
12 function requestHandler(req, res, {
13
14     var filepath = url.parse(req.url).pathname,
15         fullpath;
16
17     if (filepath === '/') {
18         filepath = 'index.html';
19     }
20
21     fullpath = path.join(__dirname, 'public', filepath);
22     stream(fullpath, res);
23 }
24
25 // read and fwd file
26 function stream(path, res) {

```

```

10 var http = http,
11 var url = url,
12 var path = path;
13 // create server
14 http.createServer(requestHandler).listen(process.env.PORT);
15
16 // handle requests
17 function requestHandler(req, res, {
18
19     var filepath = url.parse(req.url).pathname,
20         fullpath;
21
22     if (filepath === '/') {
23         filepath = 'index.html';
24     }
25
26     fullpath = path.join(__dirname, 'public', filepath);
27     stream(fullpath, res);
28 }
29
30 // read and fwd file
31 function stream(path, res) {
32
33     function fileHandler(err, data) {
34         if (err, {

```

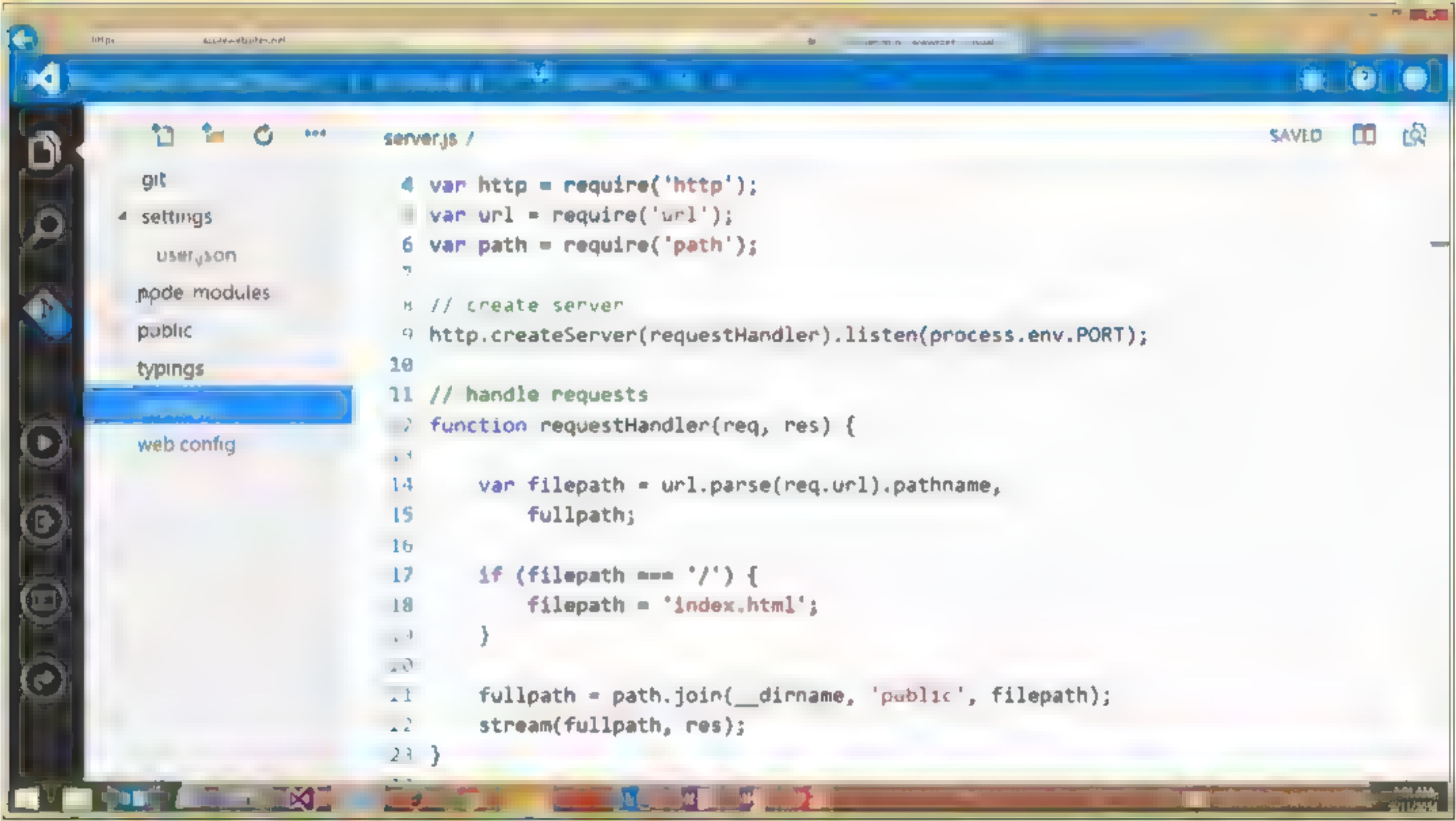
server.js

```
1      = "typings/node/node.d.ts"
2
3
4  var http = require('http');
5  var url = require('url');
6  var path = require('path');
7
8  // create server
9  http.createServer(requestHandler).listen(process.env.PORT);
10
11 // handle requests
12 function requestHandler(req, res) {
13
14     var filepath = url.parse(req.url).pathname,
15         fullpath;
16
17     if (filepath === '/') {
18         filepath = 'index.html';
19     }
20
21     fullpath = path.join( __dirname, 'public', filepath);
```

server.js

```
4 var http = require('http');
5 var url = require('url');
6 var path = require('path');
7
8 // create server
9 http.createServer(requestHandler).listen(process.env.PORT);
10
11 http.createServer(function(req, res) {
12     req.url
13 });
14
15 // handle requests
16 function requestHandler(req, res) {
17
18     var filepath = url.parse(req.url).pathname,
19         fullpath;
20
21     if (filepath === '/') {
22         filepath = 'index.html';
23     }
24 }
```





SAVED



git

settings

user.json

node\_modules

public

typings

web config

server.js /

```
4 var http = require('http');
5 var url = require('url');
6 var path = require('path');
7
8 // create server
9 http.createServer(requestHandler).listen(process.env.PORT);
10
11 // handle requests
12 function requestHandler(req, res) {
13
14     var filepath = url.parse(req.url).pathname,
15         fullpath;
16
17     if (filepath === '/') {
18         filepath = 'index.html';
19     }
20
21     fullpath = path.join(__dirname, 'public', filepath);
22     stream(fullpath, res);
23 }
```

»Chuck Norris's beard  
can type 140 wpm.«

0

»“It works on my  
machine” always holds  
true for Chuck  
Norris.«

»Chuck Norris burst  
the dot com bubble.«

main.js /

SAVED



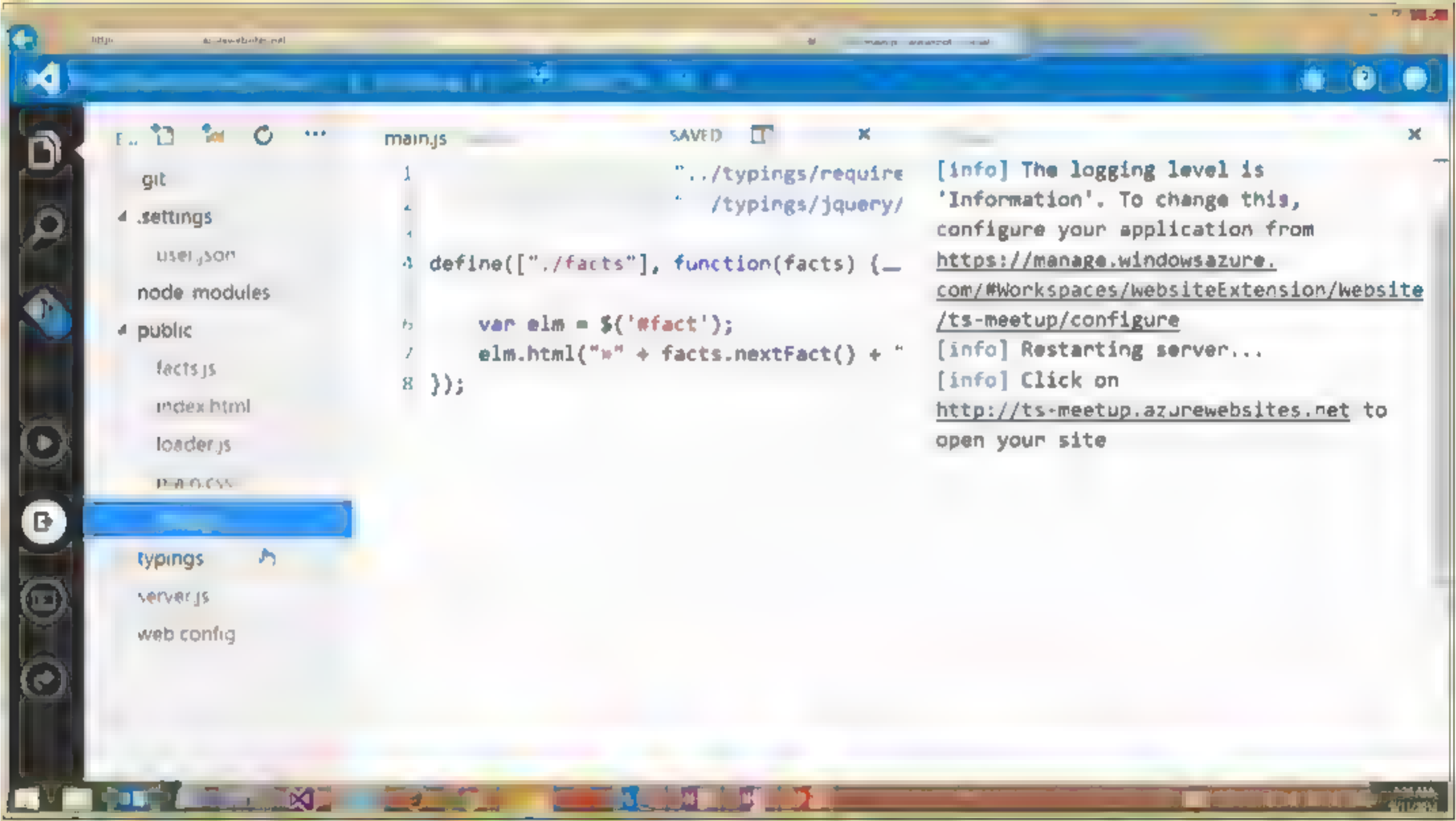
```
1  "../typings/requirejs/require.js";
2  "../typings/jquery/jquery.d.ts";
3
4  define(["./facts"], function(facts) {
5
6      $(document.body).css("background", "blue");
7
8      var elm = $('#fact');
9      elm.html(">" + facts.nextFact() + "<");
10 });
```

[info] Output will appear here from the running application at <http://ts-meetup.azurewebsites.net>

[info] The logging level is 'Information'. To change this, configure your application from <https://manage.windowsazure.com/#workspaces/WebsiteExtension/Website/ts-meetup/configure>

[info] Restarting server...

[info] Click on <http://ts-meetup.azurewebsites.net> to open your site



main.js

SAVED

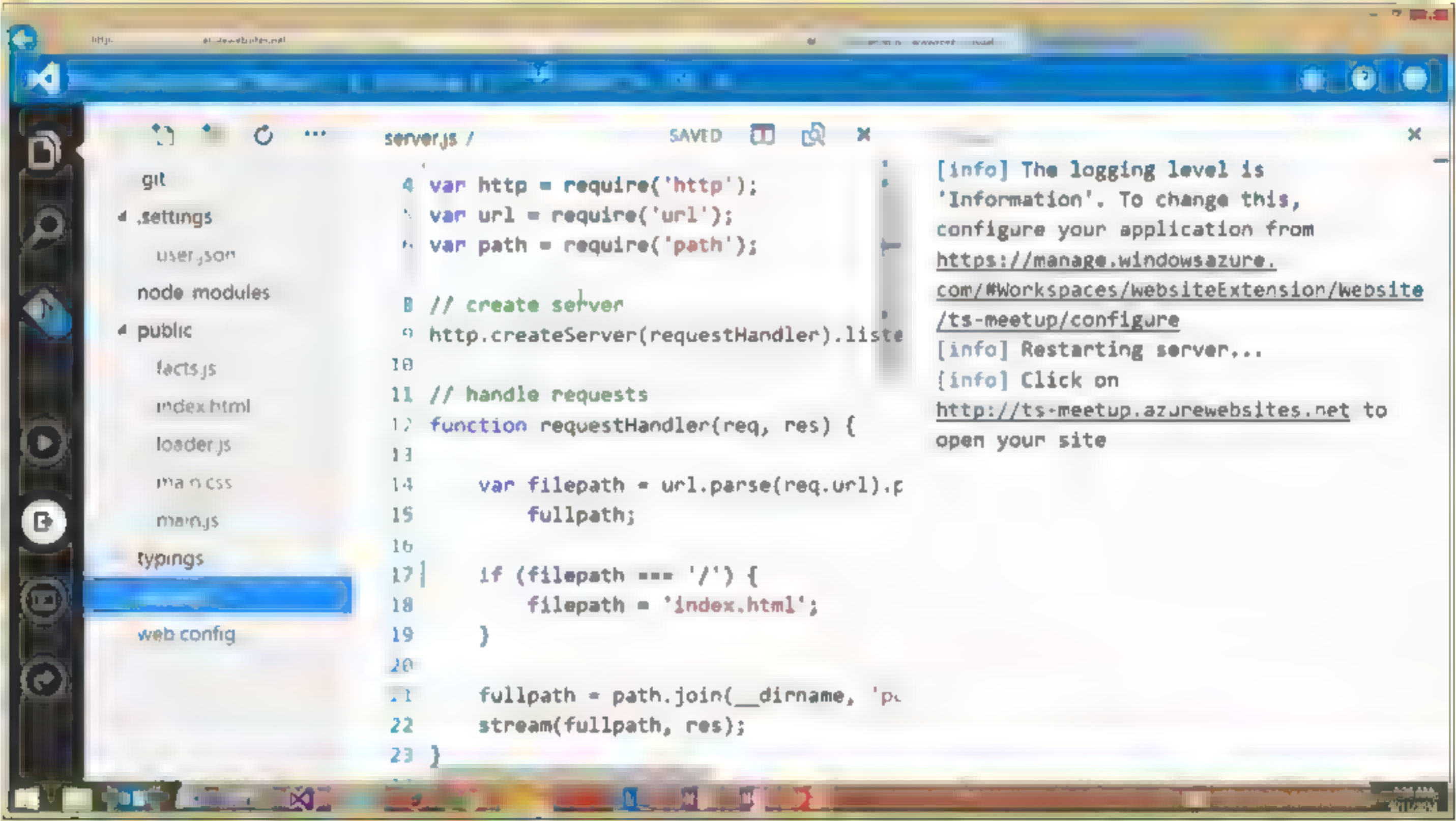
```
1      "../typings/require"
2      "/typings/jquery/"
3
4  define(["../facts"], function(facts) {
5
6      var elm = $('#fact');
7      elm.html("✖" + facts.nextFact() + "
8  });
```

[info] The logging level is 'Information'. To change this, configure your application from <https://manage.windowsazure.com/#Workspaces/websiteExtension/website/ts-meetup/configure>

[info] Restarting server...

[info] Click on <http://ts-meetup.azurewebsites.net> to open your site





server.js /

SAVED

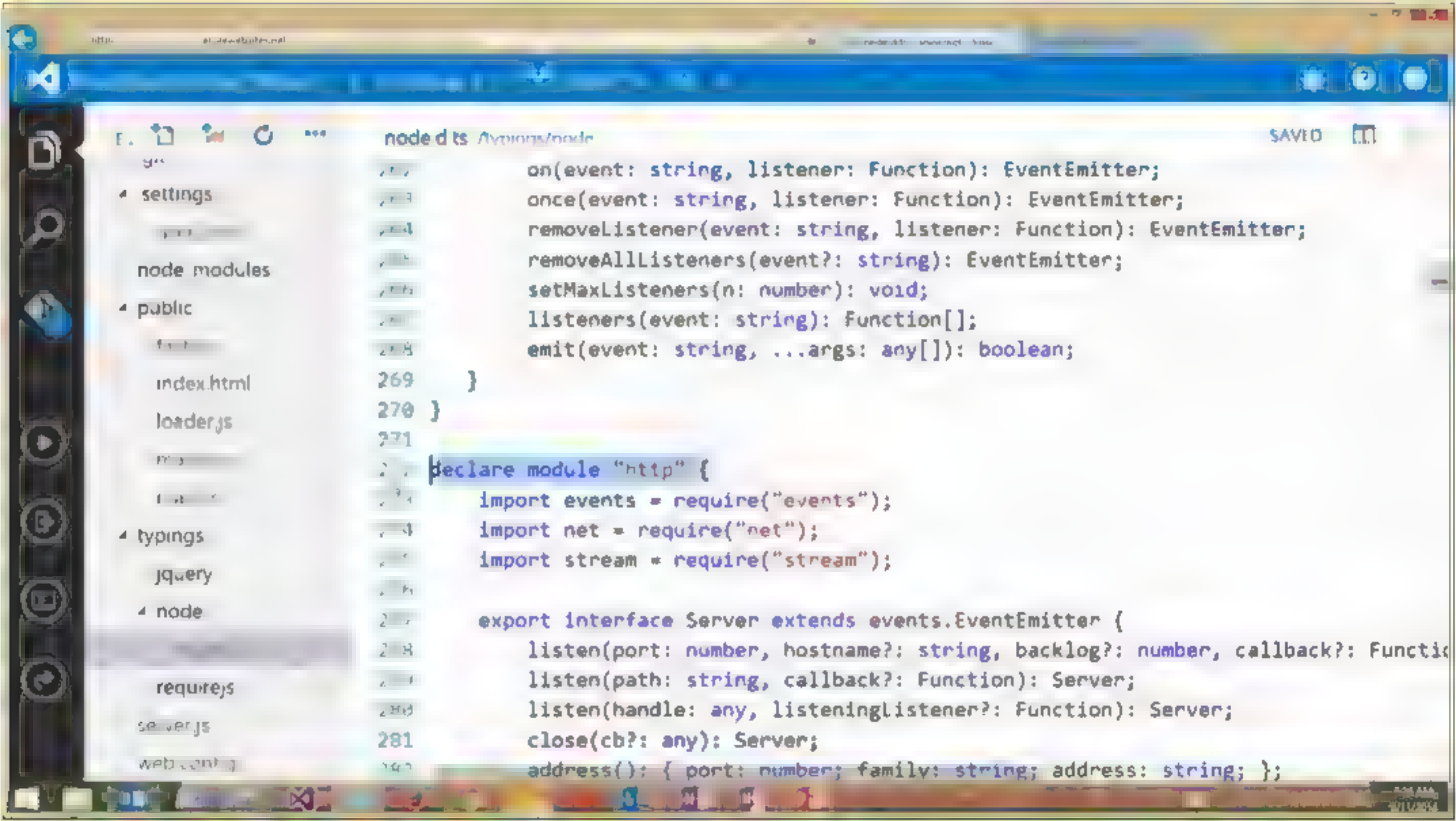
```
4 var http = require('http');
5 var url = require('url');
6 var path = require('path');

8 // create server
9 http.createServer(requestHandler).listen(
10
11 // handle requests
12 function requestHandler(req, res) {
13
14     var filepath = url.parse(req.url).pathname;
15     var fullpath = path.join(__dirname, filepath);
16
17     if (filepath === '/') {
18         filepath = 'index.html';
19     }
20
21     fullpath = path.join(__dirname, 'public', fullpath);
22     http.res.writeHead(200, { 'Content-Type': 'text/html' });
23     http.res.end(fs.readFileSync(fullpath));
24 }
```

[info] The logging level is 'Information'. To change this, configure your application from <https://manage.windowsazure.com/#Workspaces/websiteExtension/website/ts-meetup/configure>

[info] Restarting server...

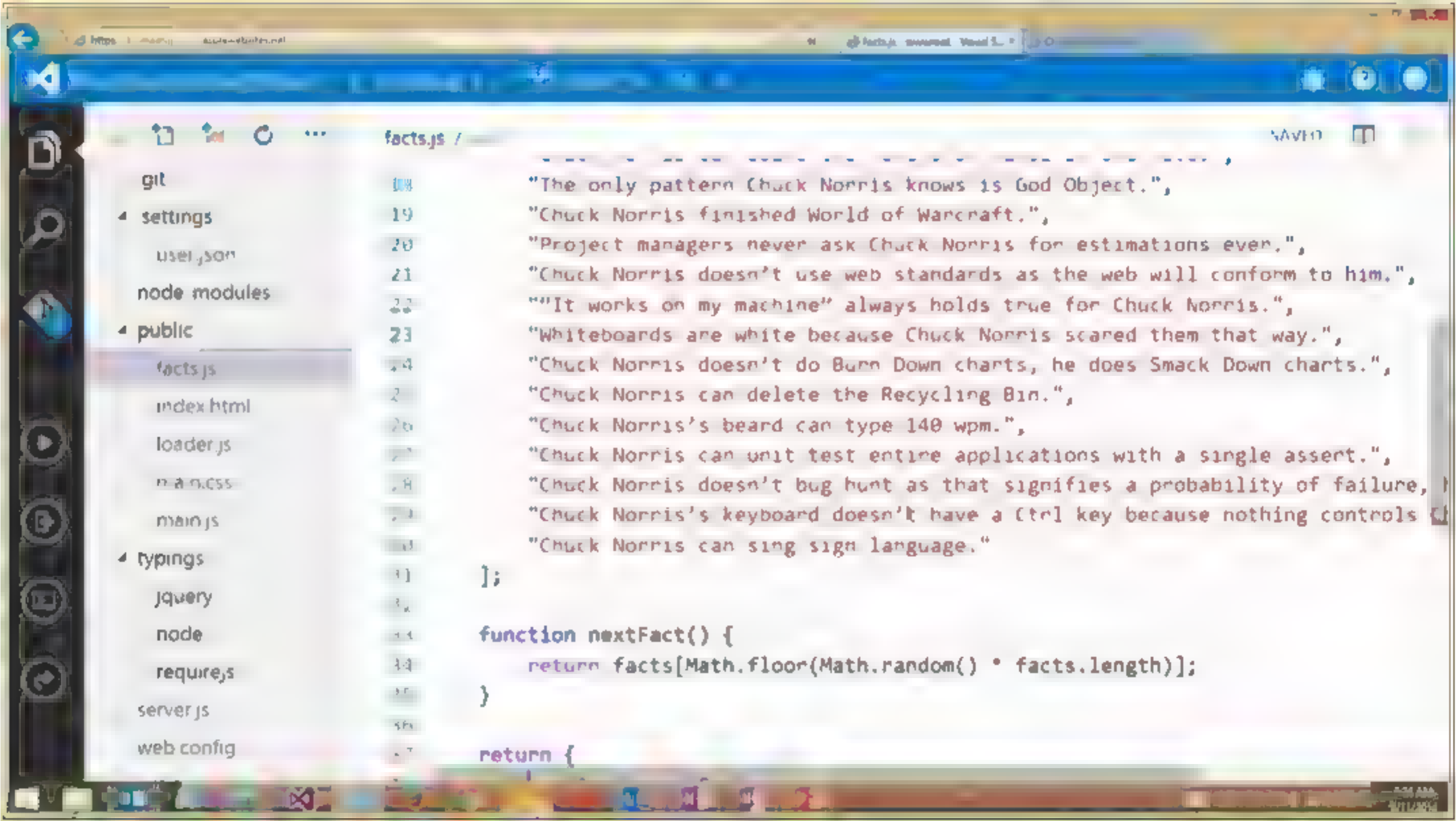
[info] Click on <http://ts-meetup.azurewebsites.net> to open your site



node.d.ts typings/node

SAVED

```
267 on(event: string, listener: Function): EventEmitter;
268 once(event: string, listener: Function): EventEmitter;
269 removeListener(event: string, listener: Function): EventEmitter;
270 removeAllListeners(event?: string): EventEmitter;
271 setMaxListeners(n: number): void;
272 listeners(event: string): Function[];
273 emit(event: string, ...args: any[]): boolean;
274 }
275 }
276
277 declare module "http" {
278     import events = require("events");
279     import net = require("net");
280     import stream = require("stream");
281
282     export interface Server extends events.EventEmitter {
283         listen(port: number, hostname?: string, backlog?: number, callback?: Function): Server;
284         listen(path: string, callback?: Function): Server;
285         listen(handle: any, listeningListener?: Function): Server;
286         close(cb?: any): Server;
287         address(): { port: number; family: string; address: string; };
288     }
289 }
```



```
1 <reference path="../../typings/requirejs/require.d.ts"
2 <reference path="../../typings/jquery/jquery.d.ts" />
3
4 define(["./facts"], function(facts) {
5
6     var elm = $('#fact');
7     elm.html(">" + facts.nextFact() + "<");
8 });
```

```
1 ../../typings/requirejs/require.d.ts"
2 <reference path="../../typings/jquery/jquery.d.ts" />
3
4 declare var exports: any; declare var module: any; declare var
5 import _var_0 = require("../facts");
6 declare function define<T>(dep: (facts: T) => T): T;
7 var _var_1 = define(["./facts"], function(facts: typeof _var_0) {
8
9     var elm = $('#fact');
10    elm.html(">" + facts.nextFact() + "<");
11 });
12 export = _var_1;
```

```

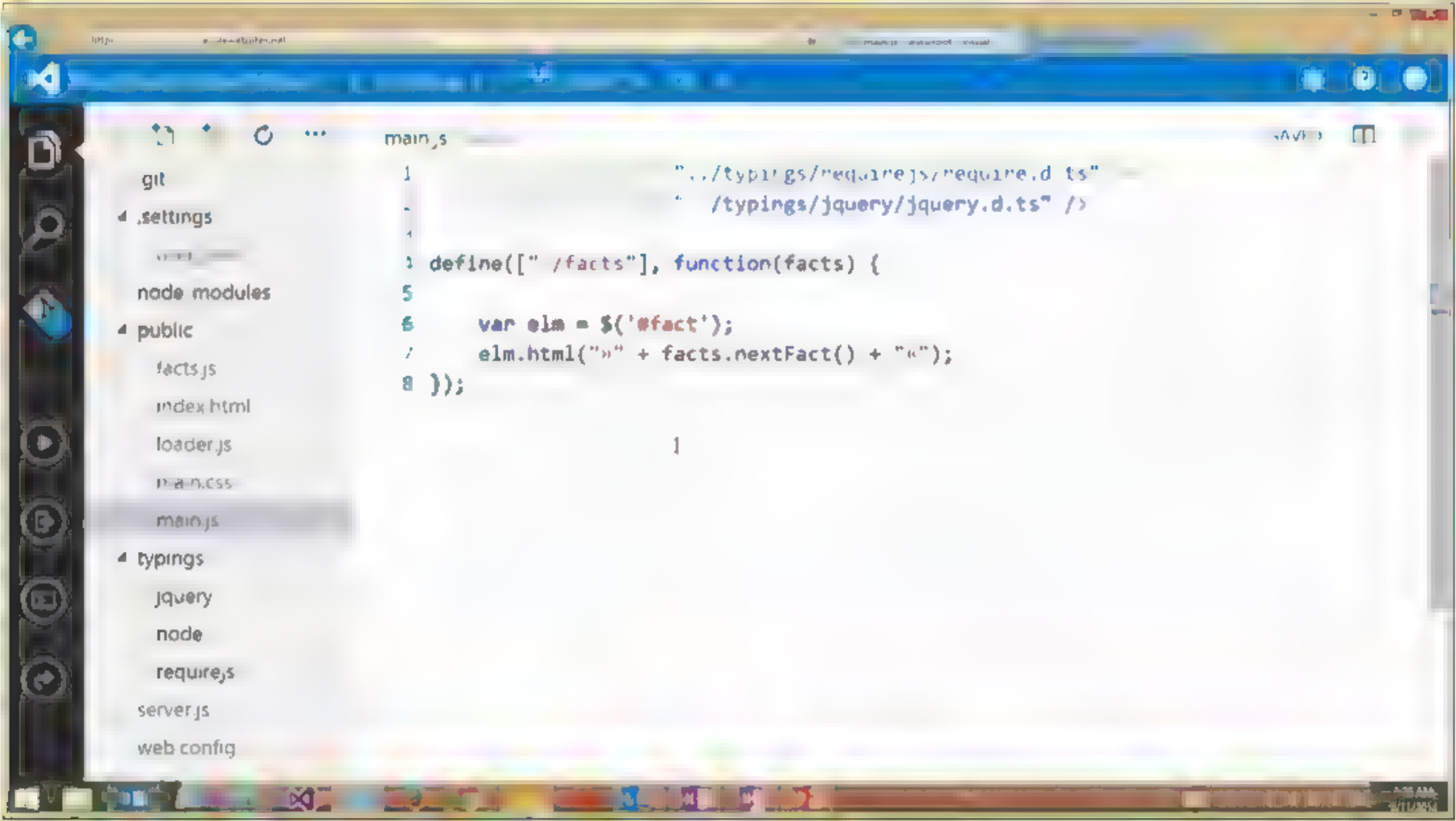
1 // @ts-ignore: 't-pings' requires 'require.d.ts'
2 // @ts-ignore: 't-pings' requires 'query.d.ts'
3
4 * define(["facts"], function(facts) {
5
6     var elm = $('#fact'),
7         elm.html('' + facts.nextFact() + ' '),
8
9     return {
10         foo: true
11     }
12
13 });

```

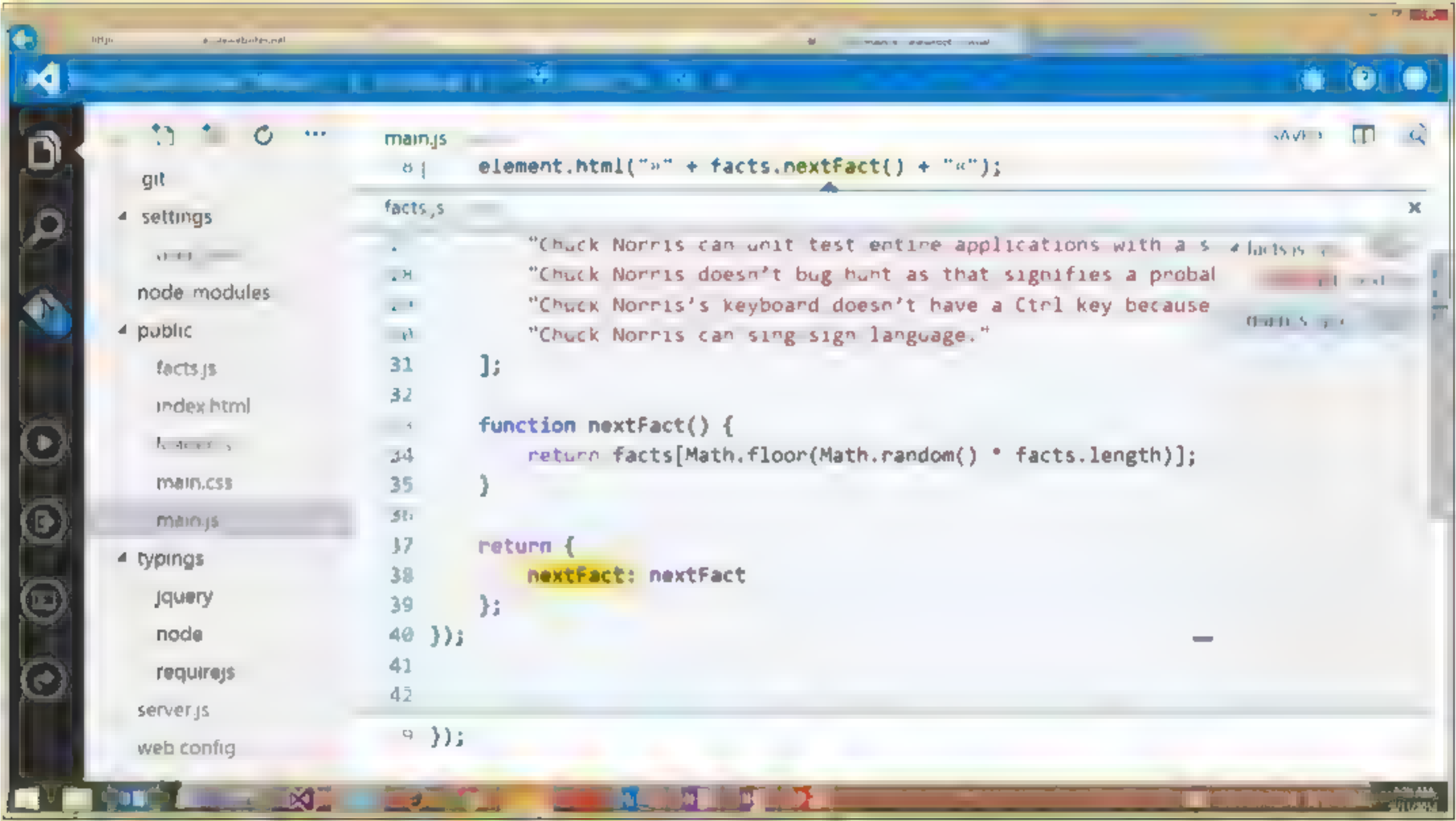
```

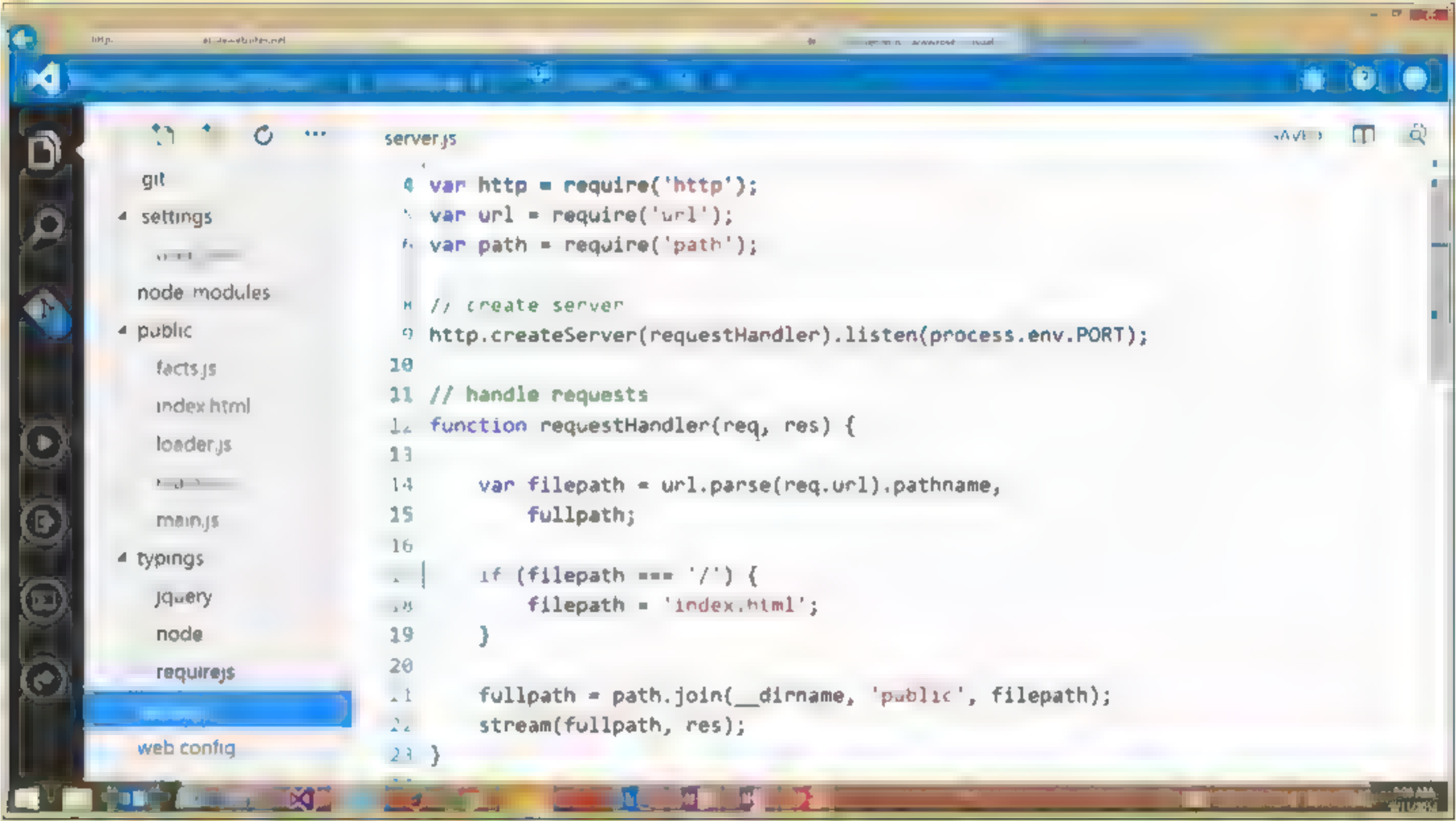
1 // @ts-ignore: 't-pings' requires 'require.d.ts' />
2 // @ts-ignore: 't-pings' requires 'query.d.ts'
3
4 * declare var exports: any, declare var module: any, declare var
5 import _var_0 = require("./facts");
6 * declare function define<T>(dep, callback (facts): T) T,
7     var _var_1 = define(["facts"], function(facts: typeof _var_0
8
9     var elm = $('#fact'),
10     elm.html('' + facts.nextFact() + ' '),
11
12     return {
13         foo: true
14     }
15
16 });
17 export = _var_1,

```









git

settings

node modules

public

facts.js

index.html

loader.js

main.js

typings

jquery

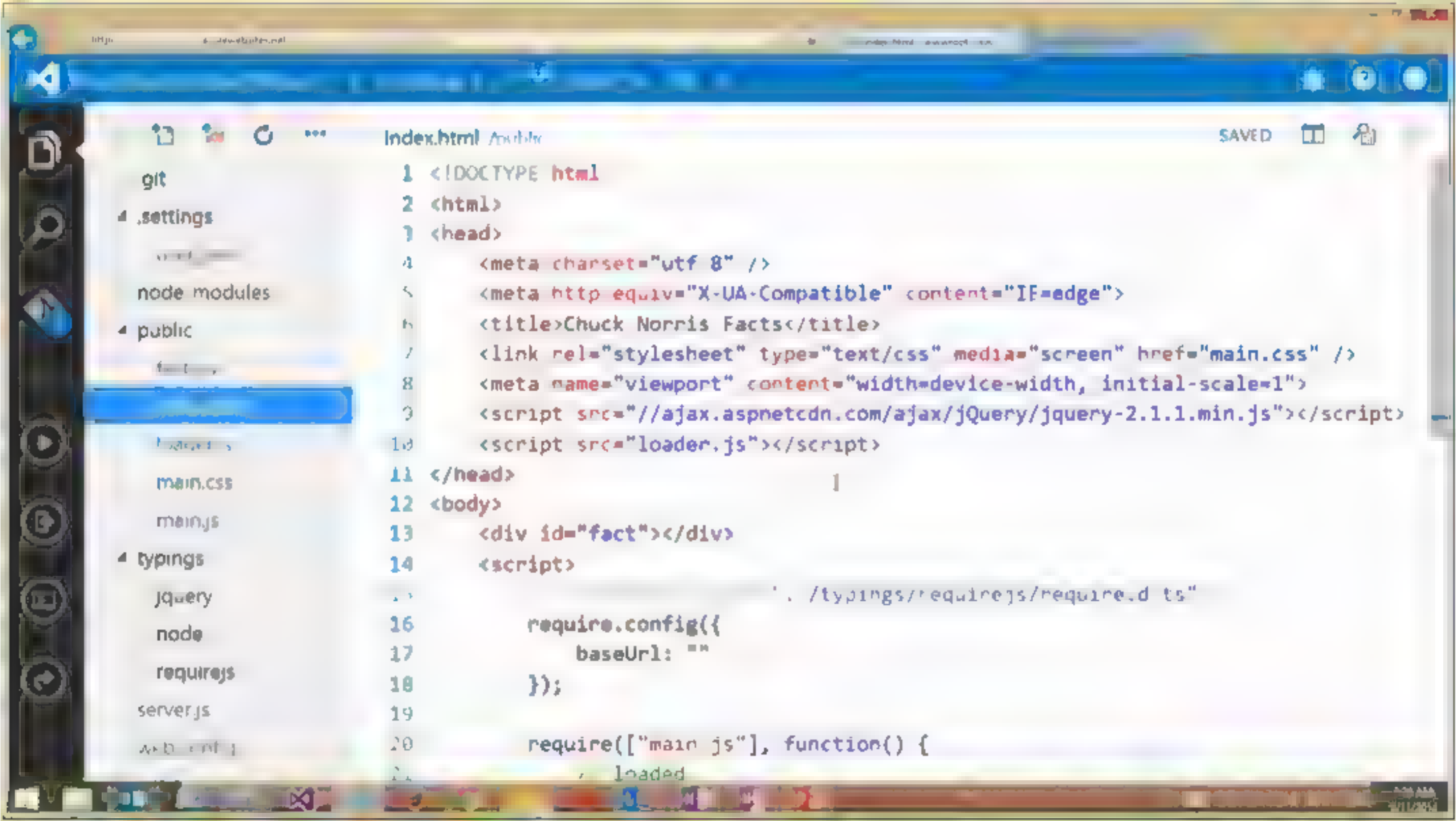
node

requires.js

web config

server.js

```
1 var http = require('http');
2 var url = require('url');
3 var path = require('path');
4
5 // create server
6 http.createServer(requestHandler).listen(process.env.PORT);
7
8 // handle requests
9 function requestHandler(req, res) {
10
11     var filepath = url.parse(req.url).pathname,
12         fullpath;
13
14     if (filepath === '/') {
15         filepath = 'index.html';
16     }
17
18     fullpath = path.join(__dirname, 'public', filepath);
19     stream(fullpath, res);
20 }
21
22 }
```



# Skype ECS UI Framework

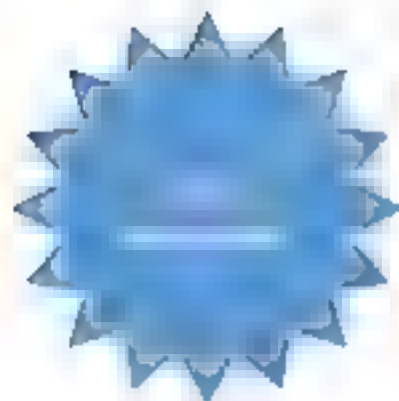
Alexander Ryzhov

# Skype ECS UI Development Approach



HTML Templates +  
attached JavaScript

- Messy HTML and JavaScript code
- Hinders reusability, promotes **copy/paste**
- Used in web sites because of SEO and legacy browsers



Pure JavaScript app



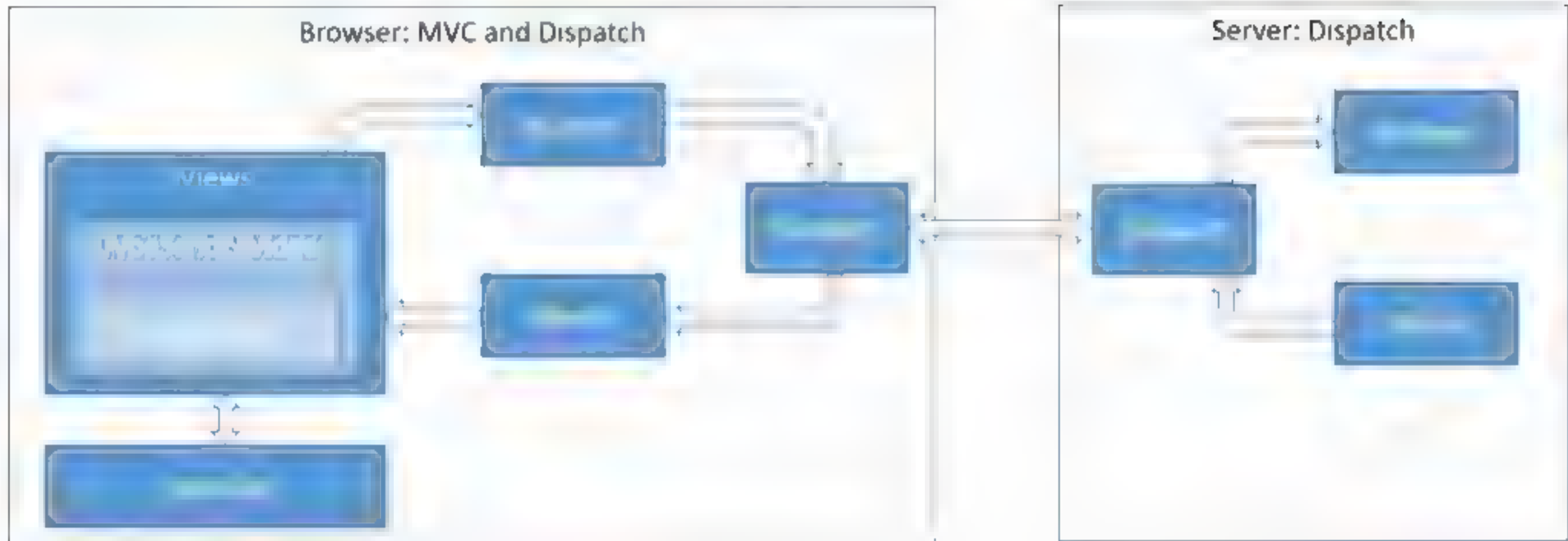
- Unlimited UI complexity
- Clean, modular code
- Requires better coding skills

# ECS UI Framework

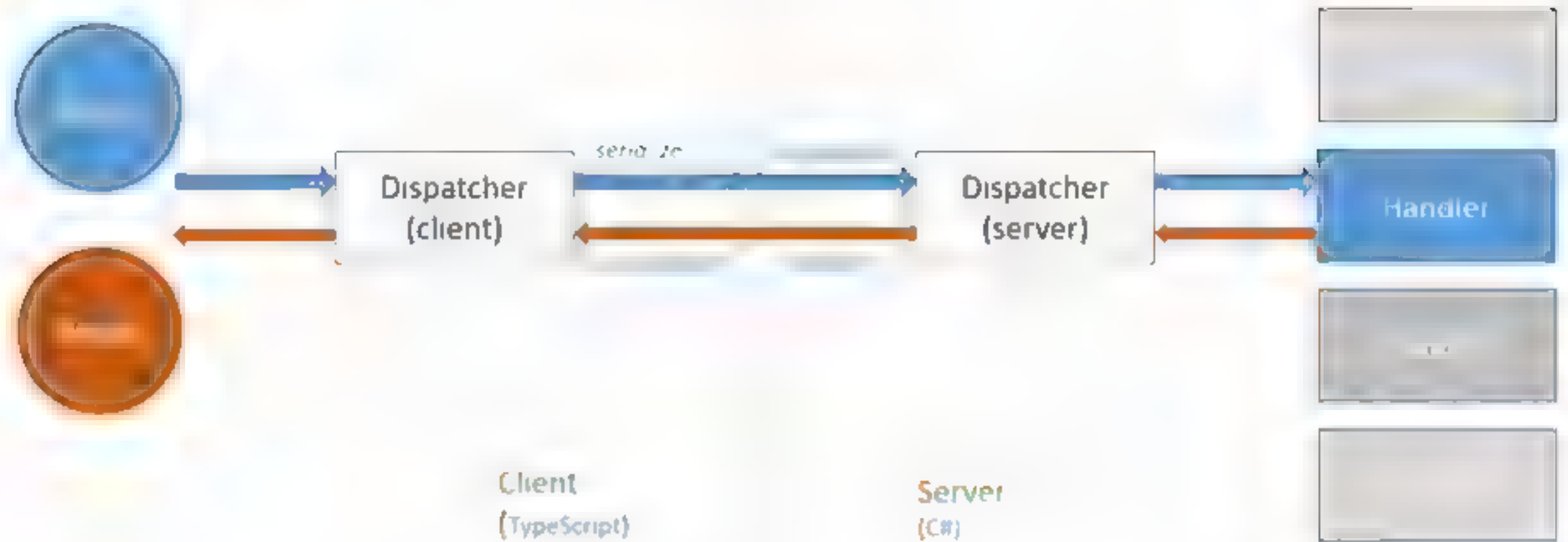
- Extensible widget library
- MVC with nested controllers and history management using #
- AJAX-based Action/Dispatcher with centralized error handling
- Forms, data binding
- CSS-based styling



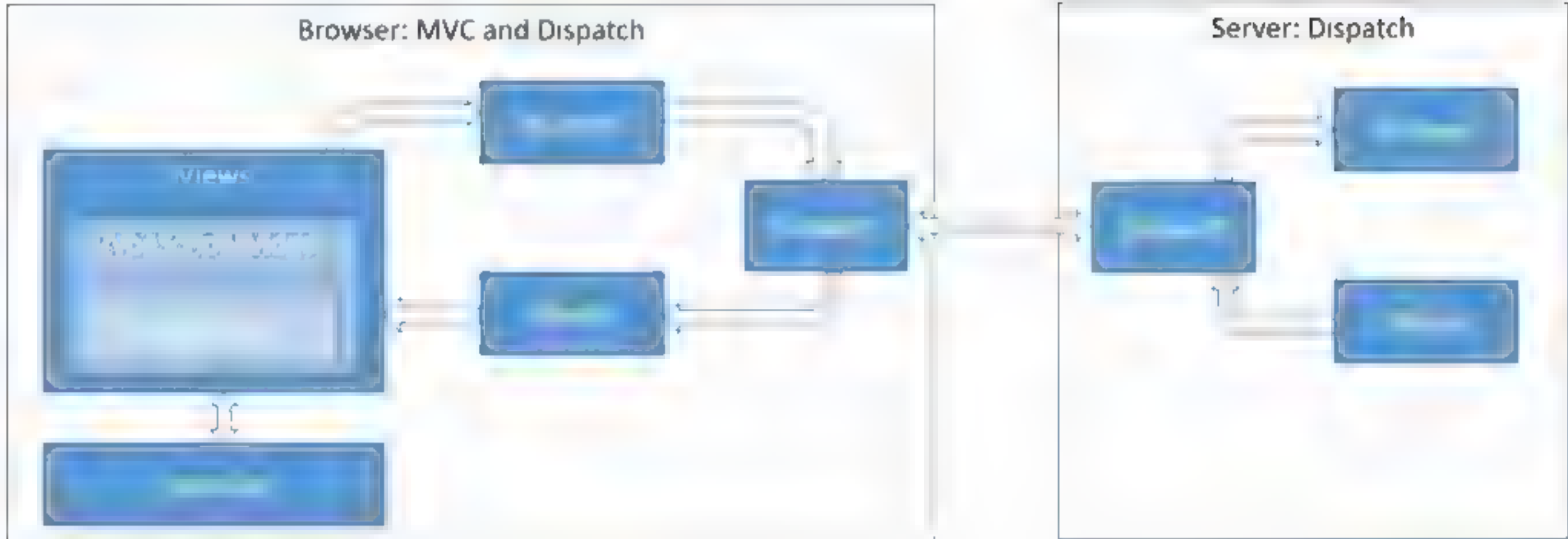
# Skype ECS UI Architecture



# Dispatcher



# Skype ECS UI Architecture



# ECS UI Framework

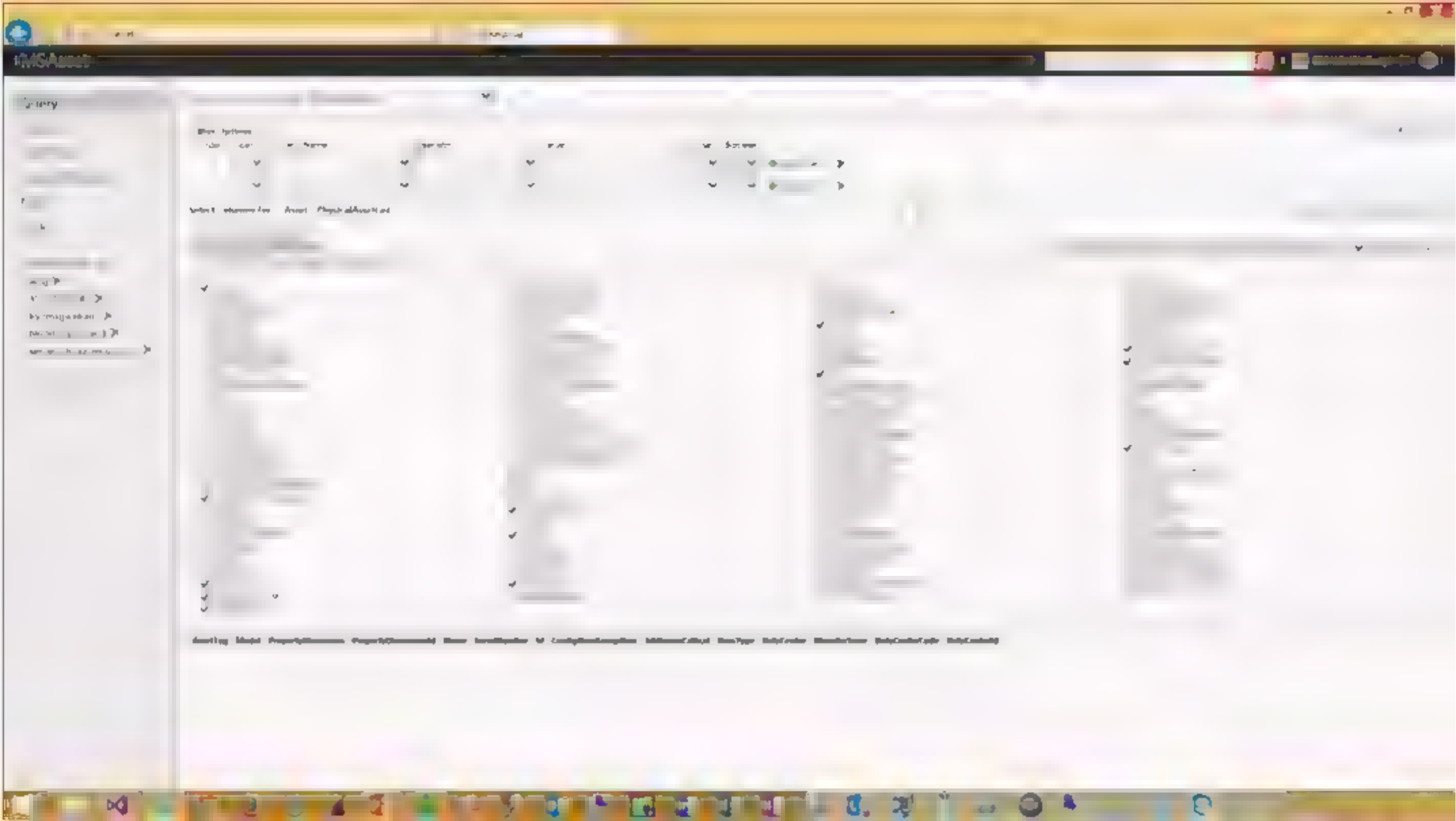
- Extensible widget library
- MVC with nested controllers and history management using #
- AJAX-based Action/Dispatcher with centralized error handling
- Forms, data binding
- CSS-based styling

Demo

# jLib Framework

- Silverlight-like programming experience
- Xaml
- Simple data binding
- MVVM pattern
- Reusable control library
- Service Reference for oData
- Linq-to-oData from UI code





# Demo

- Demo1: MSAsset – MVVM
- Demo2: Hello World!
- Demo3: Simple Binding
- Demo4: Data Binding
- Demo5: Binding to collection
- Demo6: DataGrid hierarchy
- Demo7: <http://jlib>
- Demo8: oDataGrid

# Data binding

## WPF/Silverlight

```
public class Person : IdentifyPropertyChanged
{
    private string name;
    // Declare the event
    public event PropertyChangedEventHandler PropertyChanged;

    public Person()
    {
    }

    public Person(string value)
    {
        this.name = value;
    }

    public string PersonName
    {
        get { return name; }
        set
        {
            name = value;
            // Call OnPropertyChanged whenever the property is updated
            OnPropertyChanged("PersonName");
        }
    }

    // Create the OnPropertyChanged method to raise the event
    protected void OnPropertyChanged(string name)
    {
        PropertyChangedEventHandler handler = PropertyChanged;
        if (handler != null)
        {
            handler(this, new PropertyChangedEventArgs(name));
        }
    }
}
```

## TS/jLib

```
export class Person {
    public Name: jLib.BindableProperty<string> = new jLib.BindableProperty<string>();
    constructor() {}
}
```

# oData Custom Action

## .Net/oData Proxy

```
public List<string> GetAvailableIPAddresses(
    string iprangeId,
    string startAddress,
    string endAddress,
    int pageCount)
{
    OperationParameter[] operationParameter =
        new OperationParameter[]
        {
            new BodyOperationParameter("rangeId", iprangeId),
            new BodyOperationParameter("startSearchAddress", startAddress),
            new BodyOperationParameter("endSearchAddress", endAddress),
            new BodyOperationParameter("count", pageCount)
        };

    List<string> result = this.mfxGlobalAssetModel.Execute<string>(
        new Uri(this.globalAssetSvcBaseUrl.AbsoluteUri + "/GetIPsFromIPRange"),
        "POST",
        false,
        operationParameter).ToList();

    result.Sort(CompareIPAddresses);

    return result;
}
```

## TS/jLib

```
public static GetIPsFromIPRangeAsync(rangeId: number, startSearchAddress: string, endSearchAddress: string, count: number)
```

# Markup, DataTemplate

```
<DataGrid x:Name="dataGrid" ItemsSource="{Binding HostedItems}" AllowPagination="true" PageSize="50" AllowSearchGrid="false">
  <DataGrid.ColumnTemplates>
    <!-- Name -->
    <DataGridTextColumn HeaderText="Name" Binding="{Binding Name}"/>
    <!-- ID -->
    <DataGridTemplateColumn HeaderText="ID">
      <DataGridTemplateColumn.CellTemplate>
        <DataTemplate>
          <Hyperlink Css.Name="detailslink"
                    Content="{Binding Id}"
                    NavigateUri="#"/>
        </DataTemplate>
      </DataGridTemplateColumn.CellTemplate>
    </DataGridTemplateColumn>
    <!-- Type -->
    <DataGridTextColumn HeaderText="Type" Binding="{Binding Type}"/>
  </DataGrid.ColumnTemplates>
</DataGrid>
```

- Download <http://codebox/jlib>
- Demo <http://jlib/>
- Contact [jlib@Microsoft.com](mailto:jlib@Microsoft.com)
- Hackathon Video  
<https://onehack.azurewebsites.net/project/897220bd-650c-e411-903f-00155d5066d7>
- Whitepaper <\\msgscmds\STORAGE\Users\zifengh\jLibFramework\jLibFramework.docx>